In the Claims:

1. (original) A method of operating a computing device comprising arranging for a kernel portion of an operating system for the computing device to retrieve a property published within a first process and to notify the retrieved property to one or more further processes requesting to subscribe to the property.

2. (original) A method according to claim 1 wherein the operating system is arranged to supply a retrieved property in the form of a first part comprising a property name space and a second part comprising a property type.

3. (original) A method according to claim 2 wherein the name space comprises a category part and a key part.

4. (original) A method according to claim 3 wherein the category part comprises a unique identifier (UID).

5. (currently amended) A method according to claim 3 ~~or 4~~ wherein the key part comprises a UID.

6. (currently amended) A method according to ~~any one of~~ claim[[s]] 3 ~~to~~ ~~5~~ wherein the name space comprises a 64-bit integer made of two 32-bit parts.

7. (currently amended) A method according to ~~any one of~~ claim[[s]] 2 ~~to~~ ~~6~~ wherein the property type comprises an integer value and/or a byte array descriptor.

8. (original) A method according to claim 7 wherein the byte array descriptor is of variable length.

9. (original) A method according to claim 7 wherein the integer value comprises 64.

10. (currently amended) A method according to ~~any one of~~ claim[[s]] 7 ~~to~~ ~~9~~ wherein the byte array descriptor comprises of between 0 and 512 bytes.

11. (currently amended) A method according to ~~any one of~~ claim[[s]] 7 ~~to~~ ~~10~~ wherein the byte array descriptor is provided in the form of Unicode text.

12. (currently amended) A method according to ~~any one of~~ claim[[s]] 2 ~~to 11~~ wherein the property type is provided with an access control policy defined when the property is created.

13. (original) A method according to claim 12 wherein the access control policy cannot be changed after the property has been created.

14. (currently amended) A method according to claim 12 ~~or 13~~ wherein the access control policy is defined during boot of the operating system.

15. (currently amended) A method according to ~~any one of~~ claim[[s]] 12 ~~to 14~~ wherein access control policy arranges the property in a reserved category which only allows a property to be defined in that category by a process having a write-system-data capability.

16. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel is arranged to notify to the one or more further processes only that the property has changed without specifying a new value for the retrieved property so as to enable

multiple changes in the value of the property to be notified as a single notification.

17. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel portion applies a limit on the number of further processes subscribing to the property.

18. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel portion is arranged to define the order in which the property is notified to the one or more further processes.

19. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel portion is arranged to write the retrieved property to a memory space within the computing device having a size which is predefined and not determined to the size of the retrieved property.

20. (original) A method according to claim 16 wherein the kernel portion is arranged to allocate further memory space for the writing of the retrieved property only if the retrieved property cannot be accommodated in the memory space of predefined size.

21. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel portion is arranged to use a kernel thread of known priority to notify the retrieved property to the one or more further processes.

22. (original) A method according to claim 21 wherein the kernel thread of known priority comprises a supervisor type thread of the operating system kernel.

23. (original) A method according to claim 22 comprising using a deferred function call queued on the supervisor type thread to notify the retrieved property to the one or more further processes.

24. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the property is arranged such that it can only be removed from the operating system by the process which created it.

25. (original) A method according to claim 24 wherein removal of a property from the operating system is controlled by SID.

26. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein retrieving and/or subscribing to the property is controlled by SID.

27. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the property is provided with a persistence attribute.

28. (original) A method according to claim 27 wherein the kernel portion is arranged to direct the retrieved property into persistent storage.

29. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the kernel portion is arranged to commit any outstanding change to the property to storage as part of operating system shutdown.

30. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1 wherein the property comprises a message and message queue facility for the computing device.

31. (original) A method according to claim 30 wherein the message queue is provided with a handle for enabling a message queue object to be opened by a reader and/or a writer of a message in the message queue.

32. (currently amended) A method according to claim 30 ~~or 31~~ wherein the kernel portion limits the maximum size of message that can be placed in the message queue.

33. (original) A method according to claim 32 wherein the maximum message size is 36 bytes.

34. (currently amended) A method according to ~~any one of~~ claim[[s]] 30 ~~to~~ ~~33~~ wherein the size of the message queue is fixed by a first call to open the message queue.

35. (currently amended) A method according to ~~any one of~~ claim[[s]] 30 ~~to~~ ~~34~~ wherein messages placed in a message queue are provided with a priority level for sequencing messages in the message queue.

36. (original) A method according to claim 35 wherein seven priority levels are provided for messages sequenced in the message queue.

37. (currently amended) A method according to claim 35 ~~or 36~~ wherein messages in a message queue having the same priority level are delivered from the message queue on a first in first out basis.

38. (currently amended) A method according to ~~any one of~~ claim[[s]] 30 ~~to 37~~ wherein a wait for space facility is provided for enabling, when a message queue is full when a call is made by a party to place a message on that message queue, the said message to be placed on the said message queue as soon as space becomes available on the queue without the need for a further call from that party.

39. (currently amended) A method according to ~~any one of~~ claim[[s]] 30 ~~to 38~~ wherein a wait for data facility is provided for enabling, when no messages are present on a message queue when a request to retrieve a message on the said message queue is received from a party, a message appearing on the said message queue to be notified to that party without the need for a further call from that party.

40. (currently amended) A computing device comprising an operating system arranged to cause the computing device to operate in accordance with a method as defined in ~~any one of~~ claim[[s of]] 1 ~~to 39~~.

41. (currently amended) An operating system for a computing device arranged to cause the computing device to operate in accordance with a method as defined in ~~any one of~~ claim[[s]] 1 ~~to 39~~.